call void @llvm.dbg.value(metadata ptr %a, metadata !21, metadata ...!DIExpression()),!dbg!27 call void @llvm.dbg.value(metadata ptr %b, metadata !22, metadata ...!DIExpression()),!dbg!27 call void @llvm.dbg.value(metadata ptr %c, metadata !23, metadata ...!DIExpression()),!dbg!27 call void @llvm.dbg.value(metadata i32 %n, metadata !24, metadata ...!DIExpression()),!dbg!27 tail call void asm sideeffect "dmb sy\0A\09orr x3,x3,x3\0A", "~{memory}"() ... #5, !dbg !28, !srcloc !29 tail call void asm sideeffect ".inst 0x2520e020", ""() #5, !dbg !30, !srcloc call void @llvm.dbg.value(metadata i32 0, metadata !25, metadata ...!DIExpression()), !dbg !33 %cmp11 = icmp sgt i32 %n, 0, !dbg !34br i1 %cmp11, label %for.body.preheader, label %for.cond.cleanup, !dbg !36 F for.body.preheader: %wide.trip.count = zext i32 %n to i64, !dbg !34 %0 = call i64 @llvm.vscale.i64(), !dbg !36 %1 = mul i64 %0, 4, !dbg !36%2 = icmp uge i64 %wide.trip.count, %1, !dbg !36 br i1 %2, label %pre.alc, label %Preheader.for.remaining.iterations, !dbg !36 pre.alc: %6 = call i64 @llvm.vscale.i64()%7 = mul i64 %6, 4%8 = call <vscale x 4 x i64> @llvm.experimental.stepvector.nxv4i64() Preheader.for.remaining.iterations: %9 = urem i64 %wide.trip.count, %7 %5 = phi i64 [0, %for.body.preheader] %total.iterations.to.be.vectorized = sub i64 %wide.trip.count, %9 br label %for.body %10 = insertelement < vscale x 4 x i64 > poison, i64 %7, i64 0%stepVector.update.values = shufflevector <vscale x 4 x i64> %10, <vscale x .. 4 x i64> poison, <vscale x 4 x i32> zeroinitializer br label %alc.header for.body: %indvars.iv = phi i64 [%indvars.iv.next, %for.inc], [%5, .. %Preheader.for.remaining.iterations] call void @llvm.dbg.value(metadata i64 %indvars.iv, metadata !25, metadata alc.header: br i1 true, label %lane.gather, label %lane.gather ... !DIExpression()), !dbg !33 %rem15 = and i64 %indvars.iv, 1, !dbg !43 %cmp1.not = icmp eq i64 %rem15, 0, !dbg !43 br i1 %cmp1.not, label %for.inc, label %if.then, !dbg !46 F if.then: %arrayidx = getelementptr inbounds i32, ptr %a, i64 %indvars.iv, !dbg !47 %3 = load i32, ptr %arrayidx, align 4, !dbg !47, !tbaa !49 %arrayidx3 = getelementptr inbounds i32, ptr %b, i64 %indvars.iv, !dbg !53 lane.gather: %4 = load i32, ptr %arrayidx3, align 4, !dbg !53, !tbaa !49 br label %alc.applied %mul = mul nsw i32 %4, %3, !dbg !54 %arrayidx5 = getelementptr inbounds i32, ptr %c, i64 %indvars.iv, !dbg !55 store i32 %mul, ptr %arrayidx5, align 4, !dbg !56, !tbaa !49 br label %for.inc, !dbg !57 for.inc: %indvars.iv.next = add nuw nsw i64 %indvars.iv, 1, !dbg !58 call void @llvm.dbg.value(metadata i64 %indvars.iv.next, metadata !25, ... metadata !DIExpression()), !dbg !33 alc.applied: linearized: %exitcond.not = icmp eq i64 %indvars.iv.next, %wide.trip.count, !dbg !34 br label %new.latch br label %new.latch br i1 %exitcond.not, label %for.cond.cleanup.loopexit, label %for.body, !dbg ... !36, !llvm.loop !59 new.latch: br i1 true, label %alc.header, label %middel.block Τ middel.block: %condition = icmp eq i64 %9, 0 br i1 %condition, label %for.cond.cleanup.loopexit, label %pre.alc for.cond.cleanup.loopexit: br label %for.cond.cleanup, !dbg !37 for.cond.cleanup: tail call void asm sideeffect ".inst 0x2520e040", ""() #5, !dbg !37, !srcloc tail call void asm sideeffect "dmb sy\0A\09orr x4,x4,x4\0A", " \sim {memory}"() ... #5, !dbg !40, !srcloc !41 ret void, !dbg !42